

MOVING PICTURE EXPERTS GROUP PHASE 2 TRANSPORT STREAM
DATA EDITING METHOD

BACKGROUND OF THE INVENTION

5 Field of the Invention

The present invention relates to a data editing method in a case of dividing/combining a data file in a disk recording and reproducing device, which records or reproduces MPEG2 transport stream data
10 based on the Moving Picture Experts Group (hereinafter, referred to as "MPEG") standard.

Related Background Art

Nowadays, in satellite digital broadcasting and terrestrial digital broadcasting in Japan, Europe,
15 and America, for a data format for transmitting image and audio signals, MPEG2 transport stream (hereinafter, referred to as "MPEG-2 TS") defined by IEC/ISO 13818 is being used.

MPEG-2 TS is time-division multiplexed in units
20 referred to as MPEG-2 TS packets, having a fixed length of 188 bytes and carrying information data corresponding to images and audio of a broadcast program.

If, on the receiving side, the MPEG-2 TS
25 corresponding to the broadcast program can be saved as a written record or data file onto, for example, a random access disk form recording medium such as a

hard disk or an optical disk in the same information-compressed digital signal state as the sending side, then a high quality audio/visual program can be repeatedly viewed/listened to without damaging the
5 image quality or the sound quality at all when necessary, and highly responsive random access reproduction and program editing with a high degree of flexibility would be possible.

Next, description is made of a conventional
10 example employing the above-mentioned technique. Fig. 1 shows a data structure of the MPEG-2 TS recorded onto the disk form recording medium. Fig. 2 is a diagram showing file system information for managing a data file on the disk. Figs. 3A through 6C are
15 explanatory diagrams for explaining processing methods when performing a dividing/editing process for data.

First, in a case where the MPEG-2 TS data is written onto the disk as shown in Fig. 1, it is
20 recorded consecutively into logical blocks called sectors, or, in a case of insufficient consecutive unrecorded sectors, it is recorded non-consecutively on the disk. Letters A, B1, B2, C1 and C2 of Fig. 2 refer to data recorded on the disk. Note that, in
25 this specification, the size of 1 sector is assumed to be 2048 bytes.

Up to now, as specifications for file systems

such as a File Allocation Table (hereinafter,
referred to as "FAT") and a Universal Disk Format
(hereinafter, referred to as "UDF") which are used as
formats for managing the data files on the random
5 access disk form recording medium such as a hard disk
or an optical disk, the following three rules have
been determined.

(1) The position where the data is written must
start at the beginning of a sector.

10 (2) The data cannot stop midway through a
sector other than in the sector where the end of the
data file is present.

(3) There cannot be sector areas where
different data files overlap.

15 The specifications defined by these rules must
be followed even at the time of writing and an
editing process for the MPEG-2 TS data.

According to the above-mentioned procedure, the
MPEG-2 TS data recorded on the disk is managed as a
20 file. In order to facilitate the editing process
later on, as a method of building a user interface, a
table is used for linking a file name and information
showing the file's position on the disk, including: a
file name of the MPEG-2 TS data being managed as
25 shown in Fig. 2 (the file name can be freely defined
by the user); a start sector number showing where the
data file starts to be recorded; and a data size of

an area starting from the start sector. This method enables the user to access desired data without being aware of which data is present at which position on the disk, simply by designating the file name.

5 Next, description is made of a dividing process performed on the MPEG-2 TS data file having the data structure shown in Fig. 1 and the file system information with reference to Figs. 3A through 6C and a flowchart of Fig. 7. Step numbers of Figs. 3A
10 through 6C correspond to those of Fig. 7.

 First, a user issues a request to divide the MPEG-2 TS data recorded on the disk, a file name FILE0000, an access start sector position #N, and an access data size $188 \times n_1$, at a position $188 \times n_{div}$ bytes
15 shown in Fig. 1, which is the boundary of the TS packet, to set the area from the start of the MPEG-2 TS data to the divide position as a FILE0001, and to set the area from the divide position to the MPEG-2 TS data end as a FILE0002 (step 1 of Fig. 7, Fig.
20 3A).

 In response to this request, the data from the start of the file to the divide position is registered in the file system information as the new file name FILE0001, access start sector position #N
25 and access data size $188 \times n_{div}$ (step 2 of Fig. 10, Fig. 6B).

 Here, similarly to step 2, the data from the

divide position $188 \times n_{div}$ -bytes to the MPEG-2 TS data
end position $188 \times n_1$ is set as FILE0002, with access
start sector position at $188 \times n_{div}$ bytes and access
data size of $(188n_1 - 188 \times n_{div})$ -bytes. This is because
5 the access start position would start midway through
the sector $\#N + 1$, which is not permitted by the
specifications according to the existing file system
rules. Therefore, it is necessary to perform an
alignment adjustment in order to meet both the 188-
10 byte TS packet boundary and the 2048-byte sector
boundary.

As a method of performing the above-mentioned
alignment adjustment of the TS packet boundary and
the sector boundary, it is effective to record a TS
15 packet (a NULL packet) constituted only of dummy
information having neither image nor audio data, onto
the disk.

The NULL packet is specified in the MPEG
standard. That is, the MPEG standard specifies that
20 when the data from the MPEG-2 TS stream compressed on
the sending side is expanded on the receiving side, a
data expansion process itself is not performed when
NULL packet data is transmitted.

Description is now made of the alignment
25 adjusting method for the data after the divide
position at $188 \times n_{div}$ bytes where the NULL packet was
used. First, the following formula (1) is used to

calculate an offset value L from the divide position up to the next consecutive sector #N + 1 (step 3 of Fig. 7, Fig. 4A).

$$L = 2048 - ((188 \times n_{div}) \bmod 2048) \dots (1)$$

5 Note that the operator "mod", as in "A mod B", indicates a remainder after "A" is divided by "B".

Next, based on the offset value L calculated using the formula (1), the following formula (2) is used to calculate a number n_{NULL} of NULL packets which
10 must be added in order to align the TS packet boundary at 188 bytes and the sector boundary at 2048 bytes (step 4 of Fig. 7, Fig. 4B).

$$n_{NULL} \text{ meets: } (L + 188 \times n_{NULL}) \bmod 2048 = 0 \dots (2)$$

In order to actually write onto the disk the
15 n_{NULL} quantity of NULL packets obtained using the formula (2), the disk is searched for unrecorded sector areas #X, and $188 \times n_{NULL}$ -bytes of NULL packets are recorded from sector number #X (step 5 of Fig. 7, Fig. 4C).

20 Next, the following formula (3) is used to calculate the sector number $\#N_{DIV}$ containing the MPEG-2 TS data divide position $188 \times n_{DIV}$ to be an objective (step 6 of Fig. 7, Fig. 5A).

$$\#N_{DIV} = \#N + (188 \times n_{DIV}) / 2048 \dots (3)$$

25 One sector (assumed as 2048 bytes) from sector position $\#N_{DIV}$ obtained by the formula (3) is read into a buffer such as a random access memory, and

dummy TS packet information is replaced by 2048-L bytes of data from the read start position in memory (step 7 of Fig. 7, Fig. 5B).

Further, the following formula (4) is used to
5 calculate a sector number #Xend, which is where the last of $188 \times n_{\text{NULL}}$ bytes was recorded starting from the sector #X, which is where the dummy TS packets started being recorded on the disk (step 8 of Fig. 7, Fig. 6A).

10
$$\#Xend = \#X + (188 \times n_{\text{NULL}}) / 2048 \dots (4)$$

Further, on the sector #Xend, 1 sector (2048-bytes) of the sector data that was read into the memory at step 7 is overwritten (step 9 of Fig. 7, Fig. 6B).

After performing the above-mentioned process,
15 the following file information regarding FILE0002, which is the data after the divide position, is newly registered into the file system:

File name: FILE0002

First access start sector: #X

20 First access data size: $188 \times n_{\text{NULL}} + L$ bytes

Second access start sector: $\#N_{\text{DIV}} + 1$

Second access data size: $188 \times n_1 - (188 \times n_{\text{div}} - L)$ bytes (step 10 of Fig. 7, Fig. 6C).

The above-mentioned process ends the dividing
25 process for the MPEG-2 TS data recorded on the disk.

Description is made below of an example of a data combining process. Fig. 8 is a flowchart

showing the procedure of the data combining process. Figs. 9 through 14 show transport data recorded on the disk recording medium, and changes in the data structure on the disk due to the combining process.

- 5 Note that the step numbers of Fig. 8 and the step numbers of Figs. 9 through 14 correspond to each other.

First, as shown in Fig. 9, on the disk recording medium are the MPEG-2 TS file FILE0001
10 (access start sector N, with data size of $188 \times n1$), and FILE0002 (access start sector K, with data size of $188 \times n2$), and a process request to combine these data files has occurred (S701 of Fig. 8). Note that, in Fig. 9, the file system information before
15 combining (file name, start sector number, data size) is also shown.

Hereinafter, description is made of the editing process when combining the 2 data files to make a new data file FILE0003. This process is, for example,
20 performed by an application in the disk recording and reproducing device, in response to a request from the host computer or the like. In the description below, #N, #Nend, n1 and n2 are all integers equal to or greater than 0.

25 When the process request occurs, the application uses the following formula (5) to calculate an offset value L, from sector #N which is

the data end of the data file FILE0001 on the disk recording medium, to a sector consecutively following a position $188 \times n1$ bytes from the sector #N (S702).

$$L = 2048 - ((188 \times n1) \bmod 2048) \dots (5)$$

- 5 Further, the following formula (6) is used to calculate the sector number #Nend, which contains the data end of FILE0001, as shown in Fig. 10B.

$$\#Nend = \#N + (188 \times n1) / 2048 \dots (2)$$

- Next, as shown in Fig. 11, 2048-L bytes of sector data of sector #Nend from formula (6) are read into the buffer such as a random access memory (not shown), and a process is performed to add L-bytes of dummy TS packet data from the sector data end position that was read (S704).
- 10

- 15 Next, as shown in Fig. 12A, the 2048-byte sector data in which the dummy TS packet information is added is overwritten on the sector #Nend (S705). As shown in Fig. 12B, the access start sector #N, the access size of $188 \times n1 + L$ are renewed in the
- 20 information concerning the FILE0003 that is produced in the file system after combining. The information of file FILE0003 may be renewed in the memory, or may be written onto the disk.

- Next, the following formula (7) is used to
- 25 calculate a data size M of dummy TS packet data overflowing beyond the 188-byte TS packet unit, when L bytes of the dummy TS packet data is added at S704,

as shown in Fig. 12C (S707). In other words, one sector is 2048 bytes and the dummy TS packet data is a 188-byte unit, so that the portion left over from the 188-byte dummy TS packet data at the sector end
5 as shown in Fig. 12A is calculated as a remainder M.

$$M = 188 - (L \bmod 188) \dots (7)$$

Further, as shown in Fig. 13A, the value M is used in the following formula (8), to calculate a number n_{NULL} of dummy TS packets which must be added for the
10 alignment adjustment of the sector boundary and the TS packet boundary (S708).

$$n_{\text{NULL}} \text{ meets: } (M + 188 \times n_{\text{NULL}}) \bmod 2048 = 0 \dots (8)$$

Next, after searching for the unrecorded sector area (start sector number #X) on the disk recording
15 medium as shown in Fig. 13B, the dummy data M overflowing from the TS packet boundary calculated by the formula (7), and the dummy TS packets n_{NULL} calculated by the formula (8) is written (S709).

Further, as shown in Fig. 13C, in the file
20 system information pertaining to the FILE0003, the start sector number #X and the data size $188 \times n_{\text{NULL}} + M$ bytes of information are added/renewed (S710).

Finally, as shown in Fig. 14, the file system information from the original FILE0002, its start
25 sector number #K and its access data size of $188 \times n_2$ bytes are registered into the file system information of FILE0003, and thus the combining process ends

(S711). As a result of the above-mentioned process, the FILE0003 is recognized as a combination of $188 \times n_1 + L$ bytes of data from the sector #N, sector #X, and data of data size of $188 \times n_{\text{NULL}} + M$ bytes, and data of
5 $188 \times n_2$ bytes from sector #K.

When the above-mentioned process is performed, the combining process for the MPEG-2 TS data recorded on the disk ends.

However, in the case where the user performs
10 the above-mentioned dividing/combining editing process on the MPEG-2 data file recorded on the disk, it is necessary to perform the alignment adjustment using a data structure that meets both the 188-byte TS packet boundary serving as the MPEG-2 TS data unit,
15 and the sector boundary for recognizing the edited data as a file in the file system.

In order to achieve this, it is necessary to perform many stages of process to make large-scale updates to file system information such as the
20 position and size regarding the data recorded on the disk. This places a large burden on the CPU (Central Processing Unit) governing all the processes of the recording/reproducing and editing device.

Further, when reproducing the MPEG-2 TS data
25 file newly created by the above-mentioned editing process, if the sector position where the NULL packet data was recorded, and the sector position where the

TS packet data having the actual image and audio signals is recorded, are physically very separated on the disk, this increases the number of times to perform a head seek and seek distances up to a
5 desired reproduction access start position, and there was a possibility that seamless reproduction could not be achieved.

SUMMARY OF THE INVENTION

10 The present invention has been made in view of the above-mentioned problems, and therefore has as an object to provide a data editing method capable of alleviating a processing load on a CPU without increasing head seek times.

15 According to the present invention, there is provided a data editing method for editing MPEG-2 transport stream data recorded on a disk form recording medium having a plurality of sectors, the data editing method including:

20 detecting a common boundary position which is located before one of a divide position and a combine position in data designated, and meets a packet boundary and a sector boundary; and

performing one of a dividing process and a
25 combining process for the data at the common boundary position as a reference.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram showing MPEG-2 TS data recorded on a disk form recording medium according to a conventional example;

5 Fig. 2 is a diagram showing file system information of the MPEG-2 TS data of Fig. 1;

 Figs. 3A and 3B are diagrams for explaining a data dividing process of Fig. 1;

 Figs. 4A, 4B and 4C are diagrams for explaining
10 the data dividing process of Fig. 1;

 Figs. 5A and 5B are diagrams for explaining the data dividing process of Fig. 1;

 Figs. 6A, 6B and 6C are diagrams for explaining the data dividing process of Fig. 1;

15 Fig. 7 is a flowchart showing a conventional data dividing process;

 Fig. 8 is a flowchart showing a combining process for the MPEG-2 TS data according to the conventional example;

20 Fig. 9 is a diagram for explaining the combining process of Fig. 8;

 Figs. 10A and 10B are diagrams for explaining the combining process of Fig. 8;

 Fig. 11 is a diagram for explaining the
25 combining process of Fig. 8;

 Figs. 12A, 12B and 12C are diagrams for explaining the combining process of Fig. 8;

Figs. 13A, 13B and 13C are diagrams for explaining the combining process of Fig. 8;

Fig. 14 is a diagram for explaining the combining process of Fig. 8;

5 Fig. 15A is a diagram showing a data structure, and Fig. 15B is a diagram showing file information of the MPEG-2 TS data before the dividing process according to the present invention;

10 Fig. 16A is a diagram showing a data structure, and Fig. 16B is a diagram showing file information of the MPEG-2 TS data after the dividing process according to the present invention;

15 Fig. 17 is a flowchart showing a flow of a dividing process for the MPEG-2 TS data according to the present invention;

Fig. 18 is a block diagram showing an embodiment of the present invention;

20 Fig. 19 is a flowchart showing a data combining process according to an embodiment of the present invention;

Fig. 20 is a diagram for explaining the combining process of Fig. 19;

Figs. 21A and 21B are diagrams for explaining the combining process of Fig. 19; and

25 Figs. 22A and 22B are diagrams for explaining the combining process of Fig. 19.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Embodiment 1

Next, embodiments of the present invention are described in detail with reference to the drawings.

- 5 First, in a method of dividing/editing an MPEG-2 TS data file according to Embodiment 1, a dividing process is performed at a common boundary position as a reference, which meets both a sector boundary before a position where a user requested (designated)
- 10 the dividing process concerning MPEG-2 TS data recorded on a disk and a 188-byte packet boundary. An area from the above-mentioned designated position to the common boundary position (a maximum of 94 Kbytes can be edited each time) is forcibly
- 15 overwritten with a TS packet that is constituted of dummy information on which data decoding expansion is not performed, by a decoding device for decoding private data, NULL data, or other such transport stream. Note that, NULL packet data is used as a
- 20 dummy TS packet data format.

By performing the above-mentioned process, it is not necessary to write a new data record onto an unrecorded area on the disk to align the sector boundary and the TS packet boundary.

- 25 The area from the object data start position to the common boundary position is registered as a single MPEG-2 TS data file produced after the

dividing process. The area from the common boundary position to the end of object MPEG-2 TS data is registered as another MPEG-2 TS data file. Simply renewing file system information is sufficient.

- 5 Therefore, it is not necessary to go through many stages of process and greatly renew the file system information.

The above-mentioned embodiment enables convenient dividing process for the MPEG-2 TS data
10 file recorded on a disk form recording medium having an existing file system.

Fig. 15A shows a data structure of the MPEG-2 TS recorded on the disk before the dividing process. Fig. 15B shows file system information of the MPEG-2
15 TS data of Fig. 15A. Fig. 16A shows a data structure on the disk after the dividing process for the MPEG-2 TS file according to this embodiment. Fig. 16B shows file system information after the dividing process for the MPEG-2 TS data of Fig. 16A. Further, Fig. 17
20 is a flowchart showing the dividing process for the MPEG-2 TS data file according to this embodiment. Hereinafter, description is made of a data dividing process method according to this embodiment, with reference to Figs. 15A through 17.

- 25 Note that a hardware structure in this embodiment is, for example, an optical disk device for recording/reproducing information on an optical

disk, being connected to a personal computer. In response to a request from an application in the personal computer, the optical disk device performs the data dividing process, which is described below.

5 According to this embodiment, as shown in Fig. 15A, regarding a MPEG-2 TS data file FILE_A having a data size of $188 \times n_A$ bytes recorded on the disk starting at a sector #S, a user requests to divide the MPEG-2 TS file at a position $188 \times n_{div}$, which is a
10 packet boundary of the 188-byte TS packet, and make two files. One is a FILE_B having the data from the data position head to the divide position $188 \times n_{div}$. Another is a FILE_C having the data from the divide position to the end of the data file. Description is
15 now made of the dividing process in the case where the above-mentioned request occurs.

When the above-mentioned request occurs (step 1 in Fig. 17), first, the following formula (9) is used to calculate an offset value L from the divide
20 position $188 \times n_{div}$ requested by the user (application) up to the boundary between a sector #S_{Div}-1 located forward (i.e., in the direction opposite the reproduction direction) from the divide position (step 2).

25
$$L = (188 \times n_{div}) \bmod 2048 \dots (9)$$

Next, from the offset value L, the following formula (10) is used to calculate the TS packet data size M

extending beyond the area from the sector $\#S_{DIV}$ to the sector $\#S_{DIV}-1$ (step 3).

$$M=188-(L \bmod 188) \dots (10)$$

Next, based on M which was obtained using the formula
5 (10), the following formula (11) is used to calculate a number n_{appl} of TS packets existing from the sector boundary before the sector $\#S_{DIV}$ up to the position at which the sector boundary meets the TS packet boundary (step 4).

10 $n_{appl} \text{ meets: } (188 \times n_{appl} + M) \bmod 2048 = 0 \dots (11)$

Further, when the number of packets is obtained, the following formula (12) is used to calculate a sector position $\#S_{aligned}$ which meets the alignment of both the sector boundary before the divide position
15 $188 \times n_{div}$ bytes requested by the user and the TS packet boundary (step 5).

$$S_{aligned} = 188 \times n_{div} - (L + M \bmod 188 \times n_{appl}) \dots (12)$$

Based on the value obtained above, the MPEG-2 TS data file $FILA_A$ recorded on the disk is
20 overwritten with $(L + M + 188 \times n_{appl})$ -bytes of TS packet data as NULL packets, actually on the disk, starting from the sector $\#S_{aligned}$, as shown in Fig. 16A (step 6).

When the above-mentioned process for
25 overwriting the NULL packets is completed, as shown in Fig. 16B, the MPEG-2 TS data having the data size from the access start sector $\#S$ of the MPEG-2 TS data

FILE_A before division up to the sector #S_{aligned} where the NULL packet overwriting is started, is set as FILE_B. The MPEG-2 TS data starting access from #S_{aligned} and having a data size up to the end of the data of FILE_A is set as FILE_C. These files FILE_B and FILE_C are registered into the file system information (step 7).

By performing the above-mentioned process, the dividing process that is performed on the MPEG-2 TS data file recorded on the disk according to this embodiment is completed.

Note that each time the dividing/editing process is performed on one of the MPEG-2 TS data, the data size to overwrite by the NULL packet is at most 94 Kbytes, which is the least common multiple of 2048 and 188. However, the MPEG-2 TS data transfer rate used in digital broadcasting is 25 Mbits per second, which is extremely fast. Therefore, it takes only approximately 0.0308 seconds, which is instantaneous for the user, to overwrite the data with the 94 Kbytes of NULL packets at most. Therefore, this level does not cause visual disturbance.

Embodiment 2

Next, detailed description is made of Embodiment 2 of the present invention with reference to the drawings. Fig. 18 is a block diagram showing

an embodiment of the present invention. Fig. 19 is a flowchart showing a combining process for data according to this embodiment. Figs. 20 through 22B are diagrams for explaining each process performed during the data combining process. Note that, the step numbers of Fig. 19 correspond with the step numbers of Figs. 20 through 22B.

First, in this embodiment, as shown in Fig. 18, a superior control device 101 such as a host computer is connected to a disk recording and reproducing device 102. In response to a request from the superior control device 101, the disk recording and reproducing device 102 performs recording of the MPEG-2 TS transport stream data to a disk recording medium 100, or performs reproduction of the recorded data. As the disk recording medium 100, a hard disk, an optical disk, or the like may be used.

Further, in response to the request from the superior control device 101, the disk recording and reproducing device 102 performs an editing process such as the combining process on the file data. A CPU (Central Processing Unit) 103 is provided inside the disk recording and reproducing device 102. This CPU 103 performs control of each part of the device 102. In response to the request from the superior control device 101, the CPU 103 performs control to record/reproduce the data on the disk recording

medium 100, or an editing processing such as a file data combining process. Since a construction of the disk recording and reproducing device 102 is commonly known, detailed description is omitted.

5 Note that the present invention is not limited to this construction. The present invention can also be used in a case where the data is recorded/reproduced, or combined/edited, or undergoes a like process merely by using the disk recording and
10 reproducing device according to an instruction from the user.

 Here, according to this embodiment, as shown in Fig. 20, two MPEG-2 TS files FILE0001 (access start sector number N, data size $188 \times n1$) and FILE0002
15 (access start sector number K, data size $188 \times n2$) are recorded on the disk recording medium. A process request to combine a FILE0002 after FILE0001 and make a new data file FILE0003, is issued from the superior control device 101 (S201 of Fig. 19).

20 When this process request occurs, first, as shown in Fig. 21A, the disk recording and reproducing device 102 uses the following formula (13) to calculate the offset L from the FILE0001 data end to the previous sector (S202). The size of one sector
25 is, for example, 2048 bytes.

$$L = (188 \times n1) \bmod 2048 \dots (13)$$

Next, as shown in Fig. 21B, the following formula

(14) is used to calculate a length in forward direction from the data end of the FILE0001 to the position where the TS packet boundary and the sector boundary meet (S203).

5 n_{MATCH} meets: $(188 \times n_{\text{MATCH}} - L) \bmod 2048 = 0 \dots$

(14)

Further, as shown in Fig. 22A, based on n_{MATCH} which was obtained, the data size section of the FILE0001 in the file system information is revised to
10 $(188 \times n_1 - 188 \times n_{\text{MATCH}})$ and renewed (S204). The file system information of the original file FILE0001 remains.

Next, file system information of the file FILE0003 after combining is made (S205). Specifically,
15 as shown in Fig. 22B, the start sector numbers of the file FILE0003 are: a first access start sector number #N having data size of $(188 \times n_1 - 188 \times n_{\text{MATCH}})$; and a second access start sector number #K having a data size of $(188 \times n_2)$.

20 At this time, as shown in Fig. 22B, the position in front of the end of the file FILE0001 at $(188 \times n_{\text{MATCH}})$ is the common boundary position where the TS packet boundary and the sector boundary match each other, as described above. According to this
25 embodiment, simply by making the file system information for the new file FILE0003 after combining, the file data combining process ends.

Here, when reproducing the file FILE0003, as shown in Fig. 22B, the data behind the common boundary position (the $188 \times n_{\text{MATCH}}$ portion) in the FILE0001 is not reproduced. The size of this portion
5 is at most 94 Kbytes, which is the least common multiple of 2048 and 188. However, the MPEG-2 TS data transfer rate used in digital broadcasting is 25 Mbits per second, which is extremely fast. Therefore, the 94 Kbytes of portion at most takes only
10 approximately 0.0308 seconds, which is instantaneous for the user. This level does not cause a visual disturbance, even through it is not reproduced.